# Host-Based Intrusion Detection for UNIX Systems

Presented by: Mark Zajicek

Practical Intrusion Detection Seminar

Gaithersburg, Maryland

April 23 - 24, 1997

# Technical Acknowledgments

| | |
|---|---|
| ● Cory Cohen | FedCIRC East |
| ● Chris Murray | FedCIRC West |

# Introduction / Overview

- Intruder trends
- Recommended practices for intrusion detection
  - tools to help check for intruder activities
- Intrusion Detection Systems

# Changes in Intrusion Profile

- In 1988
  - exploiting passwords
  - exploiting known vulnerabilities
- Today
  - exploiting passwords
  - exploiting known vulnerabilities
  - exploiting protocol flaws
  - examining source files for new security flaws
  - abusing anonymous FTP, web servers, email
  - installing sniffer programs
  - IP source address spoofing
  - denial of service attacks

# Intruder Trends

- Leveraging use of currently available technologies
- Creating easy-to-use exploitation scripts
- Developing increasingly sophisticated toolkits
- Transferring expertise to novices
- Increasing impact by targeting the infrastructure

# Typical Intruder Attack

- Locate system to attack
- Gain user access
- Gain privileged access
- Cover tracks
- Install backdoor for future use
- Engage in unauthorized activity
- Attack other hosts

# Introduction / Overview

- Intruder trends

- Recommended practices for intrusion detection
  - tools to help check for intruder activities

- Intrusion Detection Systems

# Recommended Practices for ID

Verify integrity of intrusion detection tools:

- Use reliable/verified tools
- Check file system integrity

Examine system activities:

- Inspect system logs
- Review monitoring notifications
- Inspect processes

Also:

- Look for unauthorized physical intrusions
- Review reports from other sources

# Verify Intrusion Detection Tools

- Use a previously verified set of tools and programs for intrusion detection
  - why?
    - to prevent Trojan horses
  - when?
    - before using
  - how?
    - trusted sources
    - read-only media
    - signatures

# Use Read-Only Media

- Advantages
  - more reliable
  - prevents accidental destruction
- Disadvantages
  - difficult to work with
  - not useful for some tools
  - be aware of dynamic libraries

# Cryptographic Signatures/Tools

- MD5
- Snefru
- PGP
- Others
  - SHS
  - Haval
  - MD4 & MD2
  - CRC32 & CRC16

# MD5

- **Author: RSA Data Security, Inc.**
- **URL: ftp://ftp.rsa.com/pub/md5.txt (or rfc1321.txt)**

  **http://ds.internic.net/rfc/rfc1321.txt**
- **Current Version: 5.1**
- **Description:**

  **The MD5 message-digest algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest” of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest.**

# MD5 Signatures

Message-Digest algorithm

- MD5 algorithm developed by RSA Data Security
- Placed in the public domain - RFC 1321
  (Ronald Rivest, MIT Laboratory for Computer Science)
- Produces a 128-bit signature
- Designed to be fast on 32-bit machines
- “Strengthened” version of MD4
  - slightly slower
  - more “conservative” in design
  - believed to be more secure

# Md5

- Implementor: Jim Ellis <jte@cert.org>
- URL: ftp://info.cert.org/pub/tools/md5/MD5.tar.Z
- Current Version: 5.1
- Description:

“MD5.tar.Z” contains the source code files for MD5 (taken from RFC 1321) and a simple Makefile to build an “md5” executable; also includes a bug fix and man page (provided by Ric Anderson <ric@artisoft.com>).

# Md5 Examples

- **Example usage:**

```
% md5 Makefile README md5.h md5c.c
MD5 (Makefile) = 25318ba716241a96658e8b25db36ca04
MD5 (README) = 989b680886cedbabb9db1d34fc867ef4
MD5 (md5.h) = d14ccb56e8457cd654fb7975171874b4
MD5 (md5c.c) = 7699fe39377979ed9fa85275184434ab
% md5 -s"Sourcetext"
MD5 ("Sourcetext") = b48be7ee8b6857a6a993fc1a78213aba
% md5 -t
MD5 time trial. Digesting 1000 1000-byte blocks ... done
Digest = f217fb0b8599c956eaeb81611e7a8758
Time = 1 seconds
Speed = 1000000 bytes/second
%
```

# Snefru

- Implementor: Ralph C. Merkle <merkle@xerox.com>
- URL: ftp://arisia.xerox.com/pub/hash/hash2.5a/snefru.tar.Z
- Current Version: 2.5a
- Description:

“snefru” is a one-way hash function that provides authentication. It does not provide secrecy. The data on the standard input is "hashed" with a cryptographically secure one-way hash function (also known as a "message digest", "fingerprint", "Manipulation Detection Code" or "MDC"). The hash is then printed on the standard output.

# Snefru Signatures

- Developed by Xerox
- Produces 128- or 256-bit signatures
- Slower than MD5
- Four and eight pass versions available
- Two pass version was broken in 1990
- $1000 reward for breaking four pass version
- Named after a Pharaoh of ancient Egypt

# Snefru Examples

- **Example usage:**

```
% ./snefru <input.txt
 be862a6b 68b7df88 7ebe0031 9cbc4a47
% ./snefru
123456789
 6103721c cd8ad565 d68e90b0 f8906163
% ./snefru256
123456789
 4ca72639 e40e9ab9 c0c3f523 c4449b39
 11632d37 4c124d77 02192ec2 e4e0b7a3
%
```

# PGP - Pretty Good Privacy

- Author: Philip R. Zimmermann <prz@acm.org>
- URL: http://web.mit.edu/network/pgp.html
- Current Version: 2.6.2
- Description:

  PGP (Pretty Good Privacy) is a public key encryption package to protect E-mail and data files.  It lets you communicate securely with people you've never met, with no secure channels needed for prior exchange of keys.  It's well featured and fast, with sophisticated key management, digital signatures, data compression, and good ergonomic design.

# PGP Signatures

- Provides integrity
- Provides authenticity
- Uses standard PGP keys
- Signature formed from:
  - MD5 signature
  - encrypted with author’s secret key
- Signature can be
  - included in message
  - stored in another file

# PGP Examples

- **Example usage:**

```
% pgp good.asc
...
Good signature from user "John Doe <joe@host.gov>".
Signature made 1997/03/21 14:08 GMT
...
% cat good.asc
-----BEGIN PGP MESSAGE-----
Version: 2.6
iQCVAgUALmM9MspvK4P8DALVAQF5DgP8Dm5knuj7gLSRLDCDquKrDmcAee
V2+ax0BWX4XKLg4NiDDMIvvmLJpVuXeIzcbRq9wjffLKSSN4CNpZeI6QJL
Ca83ukqVdp02DPD4x167JOsaYYPn7MDMA=
=Oxzm
-----END PGP MESSAGE-----
%
```

# Other Signatures

- Secure Hash Standard (SHS)
  - NIST Standard: FIPS 180
  - based on MD4 with enhancements
  - half as fast as MD5
- Haval
  - Yuliang Zheng, Univ. of Wollongong
  - faster than MD5
  - presented at AUSCRYPT in 1992

# Weaker Signatures

- MD2
  - RSA Data Security Inc.
  - much slower than MD5
  - restrictive license
- MD4
  - RSA Data Security Inc.
  - very fast on 32-bit machines
- CRC32 & CRC16
  - Cyclic Redundancy Checks
  - fast but not very secure

# Recommended Practices

Verify integrity of tools:

- Use reliable/verified tools
- Check file system integrity

Examine system activities:

- Inspect system logs
- Review monitoring notifications
- Inspect processes

Also:

- Look for unauthorized physical intrusions
- Review reports from other sources

# Verify Filesystem Integrity

- Look for changes to programs, files, data
  - why?
    - good intrusion detection mechanism
    - protects against Trojan horses
    - improves confidence in security of system
    - detects accidental modifications
  - how? - possible approaches:
    - backups and mirroring
    - cryptographic signature databases
      - tools
  - when?

# Filesystem Backups & Mirroring

- Advantages
  - provides disaster recovery solution
  - can be more secure
- Disadvantages
  - requires more hardware
  - difficult to synchronize

# Cryptographic Signatures

- Cryptographic signature databases
  - works like signatures on tools
  - supports varied security levels
  - compares one state to another
- Generating signature databases
  - only when system has not been compromised
  - before the system is attached to the network
  - during installation

# Signature Checks - When?

Check file signatures

- Periodically...
  - locates accidental mistakes
  - improves confidence in security of system
- When intrusion suspected...
  - will confirm/deny file modifications
  - may detect intruders activities
- When compromised...
  - identifies Trojan horses/backdoors
  - locates evidence left by intruder

# Filesystem Integrity Tools

- Tripwire
- L5
- Hobgoblin
- RIACS Auditing Package
- Raudit

# Tripwire

- **Authors:** Gene Kim <gkim@cs.purdue.edu>
  Gene Spafford <spaf@cs.purdue.edu>
- **URL:**
  ftp://coast.cs.purdue.edu/pub/COAST/Tripwire/tripwire-1.2.tar.gz
- **Current Version: 1.2**
- **Description:**

  "tripwire" is a file and directory integrity checker, a utility that compares a designated set of files and directories against information stored in a previously generated database. Any differences are flagged and logged, including added or deleted entries.

# Tripwire Features

- Supports several signature algorithms
  - MD5, Snefru, SHA, Haval, MD4
  - MD2, CRC32, CRC16, user-supplied
- Allows for easy updates to database
- Supports multiple hosts from one configuration
- Supports scanning of specific directories

# Tripwire Examples

- **Example usage:**

```
% tripwire -initialize
% tripwire
/homes/genek/research/tw/src/preen.c
  st_mtime: Wed May 5 15:30:37 1993 Wed May 5 15:24:09 1993
  st_ctime: Wed May 5 15:30:37 1993 Wed May 5 15:24:09 1993
---> File: '/homes/genek/research/tw/src/preen.c
---> Update entry?  [YN(y)nh?] y
...
% tripwire -update /etc/newly.installed.file
% tripwire -i 1 -i 2
```

# L5

- **Author: Hobbit <hobbit@avian.org>**
- **URL:**
  **ftp://coast.cs.purdue.edu/pub/tools/unix/L5/L5.tgz**
- **Current Version: 1.1**
- **Description:**

  **“l5” simply walks down UNIX or DOS filesystems, like "ls -R" or "find" would, generating listings of anything it finds there. It tells you everything it can about a file's status, and adds on the MD5 hash.**

# L5 Features

- Not a complete toolkit (not as robust as Tripwire)
  - generates file listings
  - expects output to be processed by scripts
- Attempts to identify “scripts”

# L5 Output

- **Output format:**

```
/file/name//T inode mode links uid/gid size mtime extra

where:
   T        is the file type (file, device, directory, …)
   extra    varies (MD5 checksum, major & minor number, …)
```

- **Example output:**

```
% l5 /tmp /etc /dev
/tmp//D 2 43777 6 0/0 512 2e71fccc 703
/etc/passwd//F 194 100644 1 0/10 10927 2e70ea30 0LOyVbfQFCUvq
/dev/console//C 3876 20622 1 0/0 0 2e71fc14 0,0
/dev/fd0b//B 169025 60666 1 0/1 0 2e0a2ea3 16,1
/dev/rst8//C 1693 20666 1 0/1 0 2e0a2eea 18,8
%
```

# Hobgoblin

- Author: Ken Rich <kenr@cc.rochester.edu>
- URL:
  ftp://coast.cs.purdue.edu/pub/tools/unix/hobgoblin/hobgoblin.shar.Z.uu.Z
- Current Version: 1.4
- Description:

  “hobgoblin” checks a description against what exists in the physical file system.  It then writes messages on standard output when it sees any discrepancy between the description and the actual file system.  A file system perfectly in line with the description given to hobgoblin will generate no output.

# Hobgoblin Features

- Does not implicitly use signatures
- Based on a description of the "correct" state
  - requires human definition
- Able to correct the "problems" it finds
- Able to invoke external programs (e.g. MD5)
- Able to detect incorrect values for:
  - owner, group, size (smaller and larger)
  - mtime, atime, ctime, …

# Hobgoblin Examples

- **Description format:**

```
file-list :|?|! checker-name[attribute-list](action)
```

- **Description example:**

```
/core /.rhosts /*,v ! ( rm \* ) ;
/etc/passwd : user[root]( chown \@ \* ) ;
/u : user[root] group[root] mode[drwxr-sr-x] inclusive{
   joe : mode[drwx--x--x] user[joe] group[users] ;
   mary : mode[drwx--x--x] user[mary] group[users] ;
};
```

- **Execution example:**

```
% hobgoblin description-file | /bin/sh
%
```

# RIACS Auditing Package

- **Author: Matt Bishop <Matt.Bishop@dartmouth.edu>**
- **URL:**
  **ftp://coast.cs.purdue.edu/pub/tools/unix/binaudit.tar.gz**
- **Current Version: 3.1.3**
- **Description:**

  **The audit package generates a listing of the file system being audited and compares it with a previously generated listing. The differences reveal what has changed and these changes are reported.**

# RIACS Output

- **Example output:**

```
% binaudit
The following files have been deleted:

      file name   type  mode  links  user  group  size   checksum  date
      ---------   ----  ----  -----  ----  -----  ----   --------  ----
      /etc/xyz       -  0644      1  root  net    2567  *skipped*  May  1, 1989

The following files have been changed; the previous attributes are shown on the
line with (old), and the current attributes are shown on the line with (new):

      file name   type  mode  links  user  group   size     checksum  date
      ---------   ----  ----  -----  ----  -----   ----     --------  ----
(old) /etc/inetd     -  0755      1  root  staff  29696  06586    28  Apr 13, 1989
(new) /etc/inetd     -  0755      1  root  staff  29696  06586    29  Apr 13, 1989
(old) /etc/ping      -  4755      1  root  staff  23552    *skipped*  July 9, 1989
(new) /etc/ping      -  4755      1  root  staff  23552    *skipped*  Apr 13, 1989
(old) /etc/utmp      -  0777      1  root  net     2952    *skipped*  Jul 10, 1989
(new) /etc/utmp      -  0777      1  root  net     2952    *skipped*  Jul 10, 1989
%
```

# Raudit

- **Author: Michele D. Crabb <crabb@nas.nasa.gov>**
- **URL: ftp://coast.cs.purdue.edu/pub/tools/unix/raudit.shar**
- **Current Version: 2.0**
- **Description:**

**“raudit” is a Perl script which audits each user's .rhosts file and reports on various findings. raudit will report on any entries which may be illegal.  An entry is considered illegal if the username does not match the username from the password file or if the entry contains a "+" or a "-".**

# Recommended Practices

Verify integrity of tools:

- Use reliable/verified tools
- Check file system integrity

➤ Examine system activities:

- Inspect system logs
- Review monitoring notifications
- Inspect processes

Also:

- Look for unauthorized physical intrusions
- Review reports from other sources

# Examine System Activities

- Inspect system logs for evidence of intrusions
- Review notifications from monitoring tools

  – why?
    - detect probes
    - detect intrusions
    - detect other system problems

  – when?
    - regularly / periodically
    - when intrusion suspected
    - when compromised

# What Should Be Logged?

- User activity
  - logins (failed attempts as well as successes); commands/connections by users
- Process activity
- System activity
  - reboots and shutdowns of the system
- Network connections
- All other services (mail, ftp, web servers)

# What Should I Look For?

- Absence of logs
- Obvious gaps in logs
- Obvious intrusion attempts
  - numerous failed login attempts
  - attempted logins to privileged accounts
  - repeated attempts to do something from outside your user community
- Unexplained error messages
- Unexplained system shutdowns/reboots
- Other unusual activity

# Examples

- Repeated login failures
  - from /usr/adm/messages (or equivalent)

```
Jan  1 01:23:45 myhost login: 6 LOGIN FAILURES FROM intruder.net
Jan  1 01:23:45 myhost login: 6 LOGIN FAILURES FROM intruder.net, root
```

- Web cgi-bin/phf attacks
  - from http access logs

```
123.45.678.90 - - [01/Apr/1997:01:23:45 -0500] "GET /cgi-
   bin/phf/?Qalias=x%0a/bin/cat%20/etc/passwd HTTP/1.0" 200 0
intruder.net unknown - [01/Apr/1997:21:34:56 -0500] "GET /cgi-
   bin/phf/?Qalias=x%ff/bin/cat%20/etc/passwd" - "-" 404 -
```

# Log Review Issues

- Log as much as possible (or tolerable/acceptable to your organization, based on risk)
  - – enable accounting
  - – configure syslog
- Common complaints (excuses) to avoid
  - – consumes disk space
    - disks are cheap
  - – consumes CPU cycles
    - benefits outweigh the cost
  - – time-consuming to review
    - use tools

# Log Management Tools

- Simple intrusion detection tests
  - chklastlog
  - chkwtmp
- Log monitoring tools
  - swatch
  - tklogger
  - logcheck
  - loginlog
- Other log management tools
  - trimlog
  - spar

# Chklastlog

- **Author: DFN-CERT <info@cert.dfn.de>**
- **URL:**
  **ftp://ftp.informatik.uni-hamburg.de/pub/security/dfncert/fixes/chklastlog.tar.Z**
- **Current Version: 1.0**
- **Description:**

  **“chklastlog” reads all entries from the file /var/adm/wtmp (file with information about logins and logouts) and checks for every user found in this file whether there is an entry in the file /var/adm/lastlog, too. The program will complain about userids with logins but no lastlogin information.**

# Chklastlog Examples

- Example usage:
  - With deleted lastlog entries:

```
% chklastlog
user ley deleted from lastlog!
%
```

  - Without deleted lastlog entries:

```
% chklastlog
%
```

# Chkwtmp

- **Author: DFN-CERT <info@cert.dfn.de>**
- **URL:**
  **ftp://ftp.informatik.uni-hamburg.de/pub/security/dfncert/fixes/chkwtmp.tar.Z**
- **Current Version: 1.0**
- **Description:**

  **"chkwtmp" examines the file /var/adm/wtmp for entries with no information (containing only null-bytes). If such entries are found the program prints the time window for the original entry. This is done by displaying the timestamps of the wtmp-entry before and after the deleted entry.**

# Chkwtmp Examples

- Example usage:
  - With deleted wtmp entries:

```
% chkwtmp
1 deletion(s) between Thu Sep 29 08:23:57 1994
and Thu Sep 29 14:11:58 1994
%
```

  - Without deleted wtmp entries:

```
% chkwtmp
%
```

# Swatch

- Author: Todd Atkins <Todd.Atkins@cast.stanford.edu>
- URL:
  ftp://ftp.stanford.edu/general/security-tools/swatch/swatch.tar.gz
- Current Version: 2.2
- Description:

  "swatch" is designed to monitor system activity. Swatch requires a configuration file which contains pattern(s) to look for and action(s) to do when each pattern is found.

# Swatch Examples

- **Example configuration file:**

```
/INVALID|REPEATED/    echo=inverse,bell=3
/ruserok/             echo=bold,bell=2
/su:/                 echo=bold,mail
/panic/               echo,exec=“page_admin”
/phf/                 echo=bold,mail,write=root
/xntpd/               ignore
/xxx/                 mail=joe,pipe=mycommand
```

- **Execution example:**

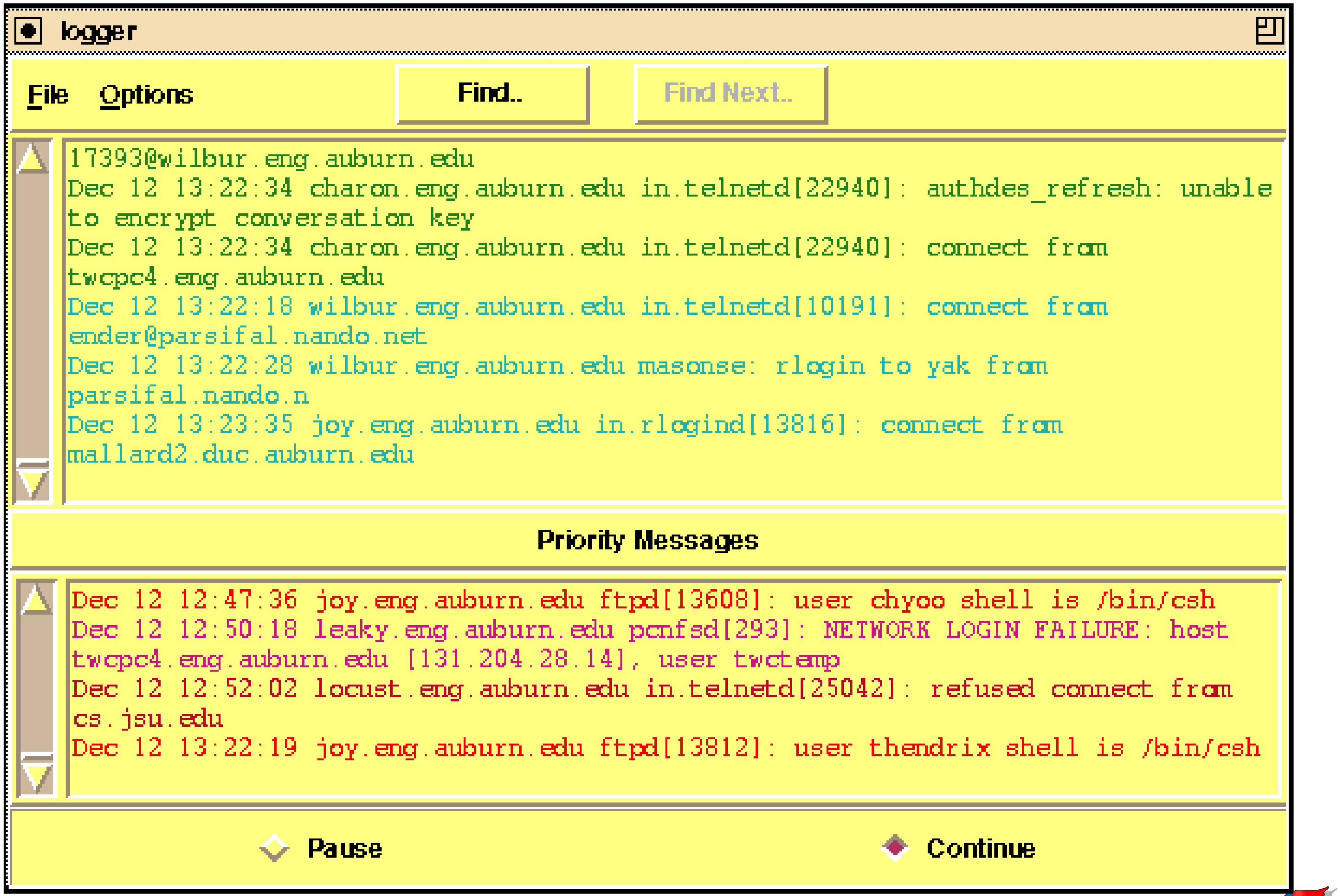
```
% swatch -c ~/.swatchrc -t /var/log/syslog
%
```

# Tklogger

- **Author: Doug Hughes <doug.hughes@eng.auburn.edu>**
- **URL:**
  **http://www.eng.auburn.edu/users/doug/second.html#TclTk**
- **Current Version: 2-beta**
- **Description:**

  **This is a program that watches log files for certain events and displays them according to certain simple rules in a priority or a normal window. It can watch multiple files and display messages in many colors.**

# Tklogger Screen



```
17393@wilbur.eng.auburn.edu
Dec 12 13:22:34 charon.eng.auburn.edu in.telnetd[22940]: authdes_refresh: unable
to encrypt conversation key
Dec 12 13:22:34 charon.eng.auburn.edu in.telnetd[22940]: connect from
twcpc4.eng.auburn.edu
Dec 12 13:22:18 wilbur.eng.auburn.edu in.telnetd[10191]: connect from
ender@parsifal.nando.net
Dec 12 13:22:28 wilbur.eng.auburn.edu masonse: rlogin to yak from
parsifal.nando.n
Dec 12 13:23:35 joy.eng.auburn.edu in.rlogind[13816]: connect from
mallard2.duc.auburn.edu
```

```
Dec 12 12:47:36 joy.eng.auburn.edu ftpd[13608]: user chyoo shell is /bin/csh
Dec 12 12:50:18 leaky.eng.auburn.edu pcnfsd[293]: NETWORK LOGIN FAILURE: host
twcpc4.eng.auburn.edu [131.204.28.14], user twctemp
Dec 12 12:52:02 locust.eng.auburn.edu in.telnetd[25042]: refused connect from
cs.jsu.edu
Dec 12 13:22:19 joy.eng.auburn.edu ftpd[13812]: user thendrix shell is /bin/csh
```

# Logcheck

- Author: Craig Rowland <crowland@psionic.com>
- URL:
  ftp://coast.cs.purdue.edu/pub/tools/unix/logcheck/logcheck-1.01.tar.gz
- Current Version: 1.01
- Description:

  "logcheck" is a software package that is designed to automatically run and check system log files for security violations and unusual activity. It is based on the frequentcheck.sh script from the Trusted Information Systems Gauntlet$^{(TM)}$ firewall package.

# Loginlog

- **Author: Mark <mark@blackplague.gmu.edu>**
- **URL:**
  **ftp://coast.cs.purdue.edu/pub/tools/unix/loginlog.c.Z**
- **Current Version: 1.0.1**
- **Description:**

  **“loginlog” runs in the background, and monitors the wtmp login file. As new entries are appended to the end of this file, messages are logged through the syslog facility recording who, when, and from where users logged into the system.**

# Trimlog

- **Author: David A. Curry** <davy@itstd.sri.com>
  <davy@vnet.ibm.com>
- **URL:**
  ftp://coast.cs.purdue.edu/pub/tools/unix/trimlog/trimlog.tar.Z
- **Current Version:** (no number)
- **Description:**

  "trimlog" is used to trim system log files to keep them from growing without bound. When invoked, it reads commands from the config file which tell it which files to trim, how to trim them, and by how much they should be trimmed. It supports trimming log files by lines and by bytes. It can send signals to processes before and after trimming their log files.

# Spar - Show Process Accounting Records

- **Author: D. Schales <Doug.Schales@net.tamu.edu>,**
  **David Hess, David Safford**
- **URL: ftp://net.tamu.edu/pub/security/TAMU/spar-1.2.tar.gz**
- **Current Version:**
- **Description:**

**"spar" is used to select records from a UNIX process accounting file. It is usually faster than most lastcomm's and significantly more flexible and powerful.**

# Recommended Practices

Verify integrity of tools:

- Use reliable/verified tools
- Check file system integrity

Examine system activities:

- Inspect system logs
- Review monitoring notifications
- Inspect processes

Also:

- Look for unauthorized physical intrusions
- Review reports from other sources

# Detecting Intruders Interactively

- Inspect processes
  - why?
    - detect intruders early to minimize damage
  - when?
    - after your log monitors alert you
  - what?
    - look for unexpected processes
    - unexpected behaviors of running processes

- Check network interface
    - unauthorized sniffers

# Process Tools

- Process monitoring tools
  - top
  - lsof

- Network interface checkers
  - ifstatus
  - cpm

# Top

- Author: William LeFebvre <wnl@groupsys.com>
- URL: ftp://ftp.groupsys.com/pub/top/top-3.4.tar.gz
- Current Version: 3.4
- Description:

  “top” is a program that will give continual reports about the state of the system, including a list of the top CPU using processes.

  Version 3 of "top" has three primary design goals: provide an accurate snapshot of the system and process state; not be one of the top processes itself; be as portable as possible.

# Top Output

- **Example usage:**

```
% top -d 1 -n 8
last pid: 19127;  load averages:  0.08,  0.19,  0.22    12:04:42
53 processes:  52 sleeping, 1 on cpu
Memory: 58M real, 1496K free, 66M swap, 104M free swap
  PID USER PRI NICE  SIZE   RES STATE   TIME   WCPU    CPU COMMAND
  347 joe    34    0   24M 8604K sleep 172:13  4.08%  4.08% Xsun
19127 joe   -25    0 1300K 1056K cpu     0:00  3.52%  3.52% top
 2681 joe     3    0   18M 9704K sleep  44:20  0.67%  0.67% netscape
  430 joe    34    0 7252K 4932K sleep   7:02  0.47%  0.47% dtwm
19023 joe    34    0 2860K 2396K sleep   2:06  0.46%  0.46% windd
  387 joe    34    0 2432K  720K sleep   1:08  0.24%  0.24% dsdm
  420 joe    34    0 3600K 1932K sleep   0:26  0.13%  0.13% ttsession
  566 joe    24    0 5780K 3576K sleep   2:35  0.10%  0.10% dtterm
%
```

# Lsof

- Author: Victor A. Abell <abe@cc.purdue.edu>
- URL: ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/lsof.tar.gz
- Current Version: 4.02
- Description:

  “lsof” lists information about files opened by processes. An open file may be a regular file, a directory, a block special file, a character special file, an executing text reference, a library, a stream or a network file (Internet socket, NFS file or UNIX domain socket). A specific file or all the files in a file system may be selected by path.

# Lsof Example

- **Example output:**

```
% lsof
inetd      98 root  5u  inet 0xf5954668      0t0  TCP *:ftp
inetd      98 root  6u  inet 0xf59545f8      0t0  TCP *:telnet
inetd      98 root  7u  inet 0xf5954588      0t0  UDP *:name
inetd      98 root  8u  inet 0xf5954518      0t0  TCP *:shell
syslogd   151 root 5uW  VREG 32,24      6 620965 /etc/syslog.pid
syslogd   151 root  4u  inet 0xf5d2a8f0      0t0  UDP *:syslog
sendmail  191 root cwd  VDIR 32,24    512  11393 /var/spool/mqueue
lsof     8684  joe txt  VREG 32,24  39888 171068 /usr/lib/libw.so.1
lsof     8684  joe txt  VREG 32,24  15720 170942 /usr/lib/libmp.so.1
lsof     8684  joe txt  VREG 32,24  15720 170935 /usr/lib/libdl.so.1
lsof     8684  joe txt  VREG 32,24 663460 171066 /usr/lib/libc.so.1
%
```

# Ifstatus

- **Author: David A. Curry <davy@ecn.purdue.edu>**
  **<davy@vnet.ibm.com>**
- **URL:**
  **ftp://ftp.cert.org/pub/tools/ifstatus/ifstatus2.0.tar.gz**
- **Current Version: 2.0**
- **Description:**

  **“ifstatus” checks all network interfaces on the system, and reports any that are in debug or promiscuous mode, which may be a sign of unauthorized access to the system. If the -v option is specified, ifstatus will print the name of each interface and the hexadecimal representation of the interface's flags word.**

# Ifstatus Examples

- Example crontab entry:

```
00   *   *   *   * /usr/local/etc/ifstatus
```

- Execution examples:
  - without promiscuous interfaces:

```
% ifstatus
% ifstatus -v
checking interface le0... flags 0x0fffffff
%
```

  - with promiscuous interfaces:

```
% ifstatus
WARNING: MYHOST INTERFACE le0 IS IN PROMISCUOUS MODE.
WARNING: MYHOST INTERFACE le0 IS IN DEBUG MODE.
%
```

# Cpm

- **Author: CERT®/CC <cert@cert.org>**
- **URL: ftp://ftp.cert.org/pub/tools/cpm/cpm.1.2.tar.gz**
- **Current Version: 1.2**
- **Description:**

**“cpm” checks whether any network interface on a host is in promiscuous mode. cpm uses standard BSD Unix socket(2) and ioctl(2) system calls to determine a system's configured network interfaces, and to check whether any of the network interfaces are currently in promiscuous mode.**

# Cpm Output

- Example output:
  - without promiscuous interfaces:

```
% cpm
2 network interfaces found:
  lo0: Normal
  le0: Normal
0 of them are in promiscuous mode.
%
```

  - with promiscuous interfaces:

```
% cpm
2 network interfaces found:
  lo0: Normal
  le0: *** IN PROMISCUOUS MODE ***
1 of them is in promiscuous mode.
%
```

# Additional Tools & Utilities

- Tools that provide enhanced logging
  - TCP wrappers
  - enhanced portmap
  - enhanced daemons
  - rfingerd
- Tools that only provide logging
  - rsucker
  - noshell
  - dummy su

# TCP Wrappers

- **Author: Wietse Venema** <wietse@wzv.win.tue.nl>
- **URL:**
  **ftp://ftp.win.tue.nl/pub/security/tcp_wrappers_7.6.tar.gz**
- **Current Version: 7.6**
- **Description:**

  **The package provides tiny daemon wrapper programs that can be installed without any changes to existing software or to existing configuration files. The wrappers report the name of the client host and of the requested service; the wrappers do not exchange information with the client or server applications, and impose no overhead on the actual conversation between the client and server applications.**

# TCP Wrappers Examples

- **Example inetd entries:**

```
tftp   dgram  udp wait root /usr/etc/tcpd in.tftpd -s /tftp
finger stream tcp nowait nobody /usr/etc/tcpd in.fingerd
```

- **Example log output:**

```
Mar 20 13:39:19 me in.telnetd[8265]: connect from x.y.edu
Mar 20 19:39:26 me in.ntalkd[9414]: connect from sam@a.com
Mar 22 18:21:13 me in.ftpd[13880]: connect from joe@x.y.edu
Mar 20 21:59:55 me in.telnetd[9742]: connect from y.z.edu
Mar 20 22:07:43 me in.telnetd[9774]: connect from w.x.com
Mar 20 23:37:51 me in.ntalkd[10021]: connect from x.y.com
Mar 22 01:39:18 me in.ftpd[13880]: connect from joe@x.y.edu
```

# Enhanced Portmap

- **Author: Wietse Venema <wietse@wzv.win.tue.nl>**
- **URL:**
  **ftp://ftp.win.tue.nl/pub/security/portmap_5beta.tar.gz**
- **Current Version: 5 beta**
- **Description:**

  **This is a replacement portmapper that prevents theft of NIS (YP), NFS, and other sensitive information via the portmapper. As an option, the program supports access control and logging in the style of the tcp wrapper (log_tcp) package.**

# Enhanced Daemons

- Author: Wietse Venema <wietse@wzv.win.tue.nl>
- URL:
  ftp://ftp.win.tue.nl/pub/security/logdaemon_5.6.tar.gz
- Current Version: 5.6
- Description:

  This package includes many enhanced versions of the standard BSD daemons: rsh and rlogin daemons that log the remote username and perform logging and access control in tcp/ip wrapper style; ftpd, rexecd and login software with fascist login failure logging and with optional support for S/Key one-time passwords; ftpd and login software that supports the SecureNet card.

# Rfingerd

- **Author: James Seng <jseng@technet.sg>**
- **URL:**
  **ftp://coast.cs.purdue.edu/pub/tools/unix/rfingerd/rfingerd.tar.gz**
- **Current Version: 1.1**
- **Description:**

  **This finger daemon is written in perl to do additional logging into a file called /var/log/trap/fingerd. It contains additional information like who is at the other end of the connect (via rfc931 : read authuser), who does he/she finger and any other information which is sent through the finger port.**

# Rsucker

- Author: Lionel Cons <cons@mercury.cern.ch>
- URL: ftp://coast.cs.purdue.edu:/pub/tools/unix/rsucker
- Current Version: 1.2
- Description:

  A Perl script that acts as a fake r* daemon and logs the attempt in syslog.
- Example inetd entries:

```
login stream tcp nowait root /usr/etc/rsucker rsucker login
shell stream tcp nowait root /usr/etc/rsucker rsucker sh
exec  stream tcp nowait root /usr/etc/rsucker rsucker rexec
```

# Noshell

- **Author: Michele D. Crabb <crabb@nas.nasa.gov>**
- **URL:**
  **ftp://coast.cs.purdue.edu/pub/tools/unix/noshell/src/**
- **Current Version: (no version)**
- **Description:**

  **“noshell” is used in place of a login shell. It records the remote host, the local user name, the remote host address, the tty line the user attached to, and possibly the remote login name (if available) to a system log file. It can also be configured to send email.**

# Dummy Su

- Author: Shawn F. Mckay <shawn@eddie.mit.edu>
- URL:
  ftp://coast.cs.purdue.edu/pub/tools/unix/dummy_su
- Current Version: 1.1
- Description:

  Dummy "su" program. Intended to help an intruder who does not know the system (many work from "cheat sheets") to trip alarms so the rightful sysadmin folks can charge to the rescue.

# Introduction / Overview

- Recommended practices for intrusion detection
  - tools for detecting intrusions
- ➤ Intrusion Detection Systems

# Intrusion Detection Research

- SRI International - Computer Science Laboratory (CSL)
  - http://www.csl.sri.com/intrusion.html
- Purdue University - COAST Project
  - http://www.cs.purdue.edu/coast/
- UC Davis - Computer Security Group
  - http://seclab.cs.ucdavis.edu/
- Lawrence Livermore National Laboratory - Computer Security Technology Center
  - http://ciac.llnl.gov/cstc/

# Intrusion Detection Systems

- Many based on Denning IDES model
- Profiling types
  - user profiling
  - intruder profiling
  - signature analysis
- Other classification of systems
  - anomaly detection
  - misuse detection
- Some IDS research projects have been extended to work in “limited” networked environments.

# ID Systems - Examples

- **SRI CSL - IDES/NIDES (1985-94)**
- **NSA NCSC - MIDAS (1988)**
- **Haystack Labs - Haystack/Stalker (1988-96)**
- **ASAX (1992)**
- **SAIC - CMDS (1991-94)**
- **DEC - POLYCENTER Security ID (1993-95)**
- **AXENT - OmniGuard/ITA (1996)**

**Multi-purpose tools**

- **LLNL CSTC - SPI/SPI-NET**

# IDES

- **Author: SRI Computer Science Laboratory (SRI/CSL)**
- **URL: http://www.csl.sri.com/intrusion.html**
- **Description:**

  **Intrusion-Detection Expert System (IDES); research prototype; real-time detection of security violations on single-target host systems. Adaptively “learns” what is normal for individual users, groups of users, remote hosts, and the overall system behavior. Observed behavior is flagged as a potential intrusion if it deviates significantly from the expected behavior or if it triggers a rule in the expert-system rule base.**

# NIDES

- **Author: SRI Computer Science Laboratory (SRI/CSL)**
- **URL: http://www.csl.sri.com/nides/index.html**
- **Description:**

**Next-Generation Intrusion Detection Expert System (NIDES); comprehensive intrusion-detection system that performs real-time monitoring of user activity on multiple interconnected systems. Analysis performed using rule-based signature analysis subsystem and a statistical profile-based anomaly-detection subsystem. The alarms generated by the two analysis units are screened by a resolver component, which filters and displays warnings through the NIDES host X-window interface.**

# MIDAS

- **Author: NSA's National Computer Security Center (NCSC)**
- **Description:**

**Multics Intrusion Detection and Alerting System (MIDAS); analyze data from a computer running the Multics operating system (as used by Dockmaster); fact and rule bases on a Symbolics Lisp machine. (NCSC intended to further develop MIDAS so it can process audit data from Sun and Apple systems.)**

# Haystack

- **Author: Haystack Laboratories, Inc.**
- **Description:**

**Developed for the Air Force Cryptographic Support Center; off-line batch mode (i.e., not designed to be used in a real-time environment); analyzed data from a Unisys mainframe (preprocessor); analysis performed on a Zenith (PC compatible, MS-DOS) machine.**

# Stalker

- **Author: Haystack Labs, Inc.**
- **URL: http://www.haystack.com/stalk.htm**
- **Description:**

**System auditing and security monitoring tool for Unix systems. Stalker tracks and reports user activities, misuses, and break-ins. Stalker provides ongoing monitoring and management, telling you who did what, when, and how.**

# WebStalker-Pro

- **Author: Haystack Labs, Inc.**
- **URL: http://www.haystack.com/webstalk.htm**
- **Description:**

**Software-based, automated management tool that patrols the perimeter of your Web site and protects the integrity of your Web server. WebStalker-Pro also manages and controls access to the contents of your Web site by allowing only authorized individuals to modify the content files. WebStalker-Pro catches outsiders and insiders alike who may be attempting to modify your Web site, and alerts you or kicks them off.**

# ASAX Project

- **Authors:** N. Habra, B. Le Charlier, A. Mounji, I. Mathieu
- **URL:** ftp://ftp.info.fundp.ac.be/pub/projects/asax
  http://www.info.fundp.ac.be/~cri/DOCS/asax.html
- **email:** amo@info.fundp.ac.be
- **Description:**

  Advanced Security audit trail Analysis on uniX; defines a normalized audit file format (NADF), and uses embedded rule-based language (RUSSEL), to solve intricate queries on any sequential data; analyze a variety of audit trails originating from different architectures (mainframes, workstations, PC). ASAX has also been applied to on-line intrusion detection and dynamic virus detection.

# CMDS

- **Author: Science Application International Corporation (SAIC)**
- **URL: http://www.saic.com/it/cmds/index.html**
- **email: cmds@cpqm.saic.com**
- **Description:**

**Computer Misuse Detection System (CMDS); audit reduction and analysis system; batch and real-time processing, interactive alert monitoring, and configurable attack signatures. Concurrently processes heterogeneous audit data from different types of computers and generates real-time alerts and graphical summary reports. Highly scalable from a few targets to several thousand.**

# POLYCENTER Security ID

- **Author: Digital Equipment Corporation (DEC)**
- **URL: http://www.digital.com/info/security/id.htm**
- **Description:**

**POLYCENTER Security Intrusion Detector; detect a wide range of security events including breakin attempts, the execution of unauthorized privileged programs, and network file transfers. Watch for activities involving a specific remote host, local user, or file. Counter intrusions by sending security alerts to the system manager and forcing users off the system. Watch for activities that are unacceptable outside of normal working hours. File daily and weekly summary reports.**

# OmniGuard/ITA

- **Author: AXENT Technologies, Inc.**
- **URL: http://www.axent.com/product/ita/ita.htm**
- **Description:**

**OmniGuard/Intruder Alert® ; Monitors system security, detects suspicious action as well as patterns of abuse, and responds automatically according to established security policy.**

# SPI-NET

- **Author: Lawrence Livermore National Laboratory Computer Security Technology Center (CSTC)**
- **URL: http://ciac.llnl.gov/cstc/spi/spinet.html**
- **Description:**

**The Security Profile Inspector for Networks (SPI-NET) supports multi-host system security inspections managed from a designated "command host." These inspections include access control testing, system file authentication, file system change detection, password testing, and checks for a variety of common system vulnerabilities. All SPI-NET command and data traffic is protected by public key encryption techniques.**

# Recommended Practices

Verify integrity of tools:

- Use reliable/verified tools
- Check file system integrity

Examine system activities:

- Inspect system logs
- Review monitoring notifications
- Inspect processes

Also:

- Look for unauthorized physical intrusions
- Review reports from other sources

# Physical Security

- Look for unauthorized hardware attached to your network.
  - why?
    - intruders may have physical access to your hardware
    - provides secure foundation
  - how?
    - put console in locked rooms
    - inspect machines for unauthorized hardware components (modems, etc.)
- Look for signs of unauthorized access to physical resources.
  - backup tapes

# Review Other Sources of Reports

- Review reports by users and external contacts about suspicious system and network events and behavior.

# Conclusions

- Logs on a typical system generate too much information to capture and analyze on a regular basis.
- Available tools can help manage, reduce, and analyze logs for specific activities.
- Research into Intrusion Detection Systems has made notable contributions; techniques are being used in the development of more commercial products.

# Intrusion Detection Resources

http://ciac.llnl.gov/ciac/ToolsUnixSysMon.html

http://cs-www.ncsl.nist.gov/tools/tools.htm

http://doe-is.llnl.gov/nitb/ids.html

http://fedcirc.llnl.gov/software/

http://seclab.cs.ucdavis.edu/

http://www.cs.purdue.edu/coast/intrusion-detection/

http://www.engarde.com/~mcn/response/spot.html

- **Intrusion Detection Systems mailing list**

  majordomo@uow.edu.au
  subscribe ids

# UNIX System Administration Books

- **Practical UNIX and Internet Security**

  **ISBN: 1-56592-148-8**
  **Authors: Simson Garfinkel and Gene Spafford**
- **UNIX System Administration Handbook**

  **ISBN: 0-13-151051-7**
  **Authors: E. Nemeth, G. Snyder, S. Seebass, T. R. Hein**
- **UNIX in a Nutshell**

  **ISBN: 1-56592-001-5**
  **Authors: Daniel Gilly & the Staff of O'Reilly & Associates**
- **Essential System Administration**

  **ISBN: 1-56592-127-5**
  **Author: Æleen Frisch**